Preliminary Copy

A Graphical User Interface to Prepare the Standard Initialization for WRF

NOAA Technical Memorandum
To Be Published July 2003

Paula McCaslin, John Smart, Brent Shaw, Paul Schultz, Steve Albers, Dan Birkenheuer, John McGinley and Linda Wharton
NOAA/Forecast System Laboratory
Boulder, Colorado

1.0 Introduction

A graphical user interface (GUI) is a commonly used computer tool that allows a user to more easily interact with underlying software and files. While the GUI itself can be quite involved and contain several levels of interactivity, it ultimately allows a safer and more efficient method to manipulate the software and modify files. Users do not have to remember the paths to files or executable names, and, therefore, they do not necessarily have to remember dependencies among these. The GUI also prevents users from making potentially serious mistakes or becoming confused about the intricacies of the system software.

The Weather Research and Forecasting (WRF) system is being developed with the latest scientific knowledge and computer (software and hardware) technologies. Several organizations are participating in this development and are using established software as well as writing new software as the system evolves. An important and necessary first step of WRF is to prepare the Standard Initialization (WRFSI). The WRFSI provides two mandatory parts of WRF:

1. To define and localize a three dimensional grid; hereafter this is referred to as domain localization; and,
2. To provide the initial and lateral boundary condition files by interpolating larger scale model data to the domain.

The Forecast Systems Laboratory (FSL) has developed the WRFSI system (Ref). The design of this system allows significant flexibility achieved using three primary directory structures: one for source software, one for installed executables and one for output data. These are further described in section 3.0. The purpose of this document is not to describe the details of the WRFSI since this can be obtained from other documentation (see references); on the other hand, a general understanding of the system is required and knowledge of the three-directory structure is important.

This initial step in preparing WRFSI presumes that a user has obtained and installed the software (i.e. downloaded the tape archive (tar) file from the WRF web site: http://wrf-model.org). This means the executable programs have been successfully compiled. Completing an implementation requires that users have some understanding of the WRFSI system design and interdependencies (these are further described in the WRFSI README in Appendix A). While the GUI we describe in this document does not install the WRF software, there is reason to believe that it could. Configuring and installing the WRFSI software may be a capability in a subsequent version of the WRF GUI. For now, installation is manual and instructions to complete this are detailed in the README file found in the WRFSI tar file.

The primary objective of this paper is to describe the current state of the WRF GUI and provide an understanding of its functionality. The GUI runs all WRFSI software; its components facilitate domain localization and interpolation of initial and lateral boundary condition data files.

1.1 Domain Localization

The WRFSI satisfies global domain localizations in the commonly used map projections: Lambert Conformal (both tangent and secant), Mercator, and Polar Stereographic. Some basic understanding of map projections is valuable but not mandatory. The phrase domain localization means to complete the following tasks:

- Prepare a directory structure where the WRF data will be written.
- Choose a rectangular domain on earth, with a fixed center point and projection type. (The GUI has built in criteria to assist the user in selecting a projection based on domain's the center latitude value).
- Edit the domain for the specific number of grid points, grid spacing, standard latitude and longitude values to insure the resulting map covers the desired area with this set of parameters.
- Determine the distribution of vertical levels.
- Verify that the paths to geography data (terrain, land-use, greenness fraction, albedo, soil type, and global mean temperature) locations are correct.
- Run the localization Perl scripts which, in turn, run the localization Fortran software.
- Verify that the localization is correct.

1.2 Interpolation of Data

In addition to domain localization, WRFSI also includes software that prepares large-scale model output for WRF model initialization and lateral boundary conditions.

Details of the software and how to install it are presented in section 2.0. In section 3.0 we describe the design and layout and give an example of localizing a domain over Alaska. Future work is discussed in 4.0. A brief summary is found in section 5.0.

2.0 Software

2.1 Selecting Perl/Tk

Perl is the scripting language used to configure WRF. It is a powerful software tool automating many aspects of the WRF setup. These scripts, however, have environment and file variables that need to be set prior to running them as well as setting many command-line arguments. This can be a complex task for users. The idea for a graphical user interface was an obvious solution to accomplish this task efficiently and accurately. Many GUI languages and tools were considered and evaluated. The GUI was written Perl/Tk. One key motivation for this choice was compatibility with the existing WRF scripts. Key justifications for this choice follow.

Perl is arguably the most popular scripting language in use today. It is used for a wide variety of tasks, including file processing, system administration, web programming and database connectivity. Early Perl users had to be content with command-line interfaces. But the splitting-off of the Tk widget library from the Tcl language opened a whole new world to Perl. Perl programmers could now easily create graphical interfaces for their programs using Tk's flexible and friendly widget set and, with little effort, those programs could be made to work across Windows and Unix platforms.

The relatively recent advent of the web browser would seem to have made the Tk interface obsolete. CGI programmers are almost inherently cross-platform and provide many of the same widgets as Tk (this includes Menus, Buttons, text entry fields, and so on). However, the inherent statelessness of the Web makes it difficult to write some programs for it. Perl/Tk provides a richer widget set than available to the CGI programmer. Server push and client pull try to get around some limitations, while JavaScript fills in other gaps, but the fact is, the user experience still fall short in many instances. It is for precisely this reason that Perl/Tk continues to flourish.

The Tk module gives the Perl programmer full access to the powerful Tk widget set. This rich and diverse library, like Perl itself, makes the easy things easy and the hard things possible. [LIDI02]

Perl/Tk programs are written in an object-oriented (OO) style. Perl/Tk widgets (such as Buttons) are objects that have methods we invoke to control them. Besides widgets, Perl/Tk has images, which are also objects. Each different widget belongs to a class. A widget's class defines its initial appearance and behavior, but individual widgets of the same class can be customized. As an example, we could create two Buttons that have different textual labels but are otherwise identical. The widget class constructor instances a widget. The class also defines a widget's initial behavior by creating bindings. A binding associates an event such as a button press with a callback, which is a subroutine that handles the event. [LIDI02]

Using Perl/Tk allows the user to call existing Perl subroutines directly. In contrast, choosing another language would add a layer of detail to communicate between to different languages -- a step not necessary using Perl/Tk.

Further motivations in selecting Perl/Tk is that it is free, platform independent without modification or recompiling and it is open source software that is very well supported in the community.

2.2 Other software used in GUI (Perl, C, Fortran)

As mentioned, Perl is the scripting language used to configure WRF. Many Perl scripts directly edit, modify, and create files and directories used by the WRF system and its processes. There are several Perl scripts invoked from the GUI by simply pushing a button. For example, window_domain_rt.pl is called from the GUI with all the necessary command-line arguments resolved by the GUI.

The projection mapping software is written in C. The C compiled software allows for faster code than would be possible in Perl since this is, generally, interpreted code. One approach than might be considered again in the future is to extend Perl by creating a C library that Perl can understand and link with. Forgoing this, we currently execute C code from Perl, which generates output files to /tmp that are subsequently read as input to Perl/Tk.

The WRFSI software is written mostly in Fortran, Fortran90, C, and Perl. It is fully dynamic by virtue of the Fortran namelist entries for domain size and configuration. Much of the software has roots from the FSL Local Analysis and Prediction System, (LAPS). LAPS has been in existence for many years and the grid localization software is particularly robust. It is used by many researchers, operational weather centers and organizations world-wide and has gone through numerous

2.3 Installing

The Tk extension does not come with the standard Perl distribution. If it is not on your system it is easy to install by visiting the CPAN (Comprehensive Perl Archive Network) at www.cpan.org/authors/id/NI-S, looking for the version in the form of Tk800.023.tar.gz. Once you get the tar file, unzip, untar and install it.

The WRF GUI is located with the WRFSI source code in a directory called SRCROOT/gui.

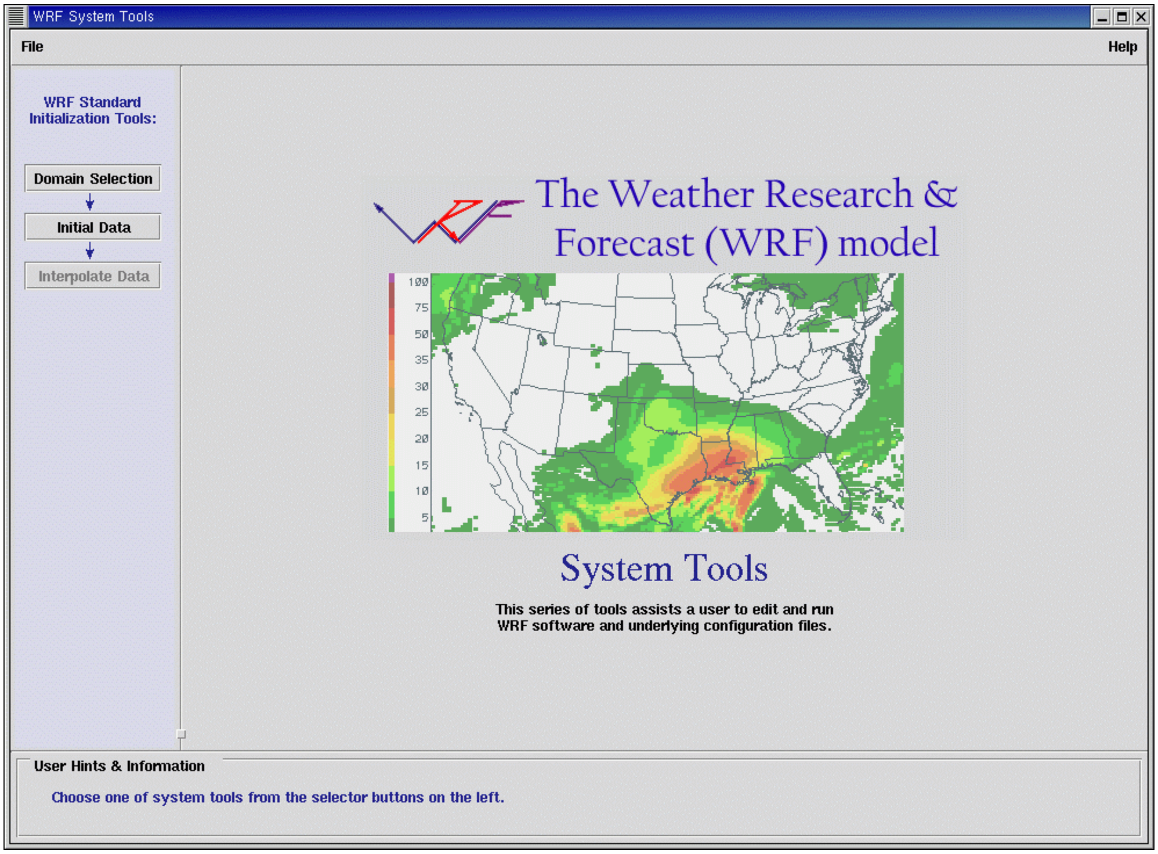

Figure 1: Display of WRF GUI application at start-up.

3.0 Layout and Design

The GUI described here is customized for WRF and a banner image is displayed upon start-up (Figure 1). The GUI automatically determines a WRF realization through a simple evaluation based upon the system design and file structure. The GUI is capable of running other systems besides WRF and, as such, a different banner would be displayed. We allow this capability because other weather analysis and forecasting systems can use this GUI to configure and localize domains (in particular, LAPS).

The WRFSI environment, in which the GUI runs, allows the source software directory to be located in one area (hereafter, SRCROOT), the executable software to be located in a second directory (hereafter, INSTALLROOT) and the I/O directory in a third area (hereafter, DATAROOT). The nature of this design allows full flexibility of the system but it also adds a degree of complexity making it difficult to set up and initially run. Experienced users are more likely to succeed on the first try; however, it is typical for users that fail on the first try to give up. The three-directory structure approach is implemented in the system by using environment variables. Because the GUI is delivered with the tar file, the SRCROOT is known (i.e. the location in which the tar file was “un-tarred”). Users run the GUI from the INSTALLROOT so this is also known. (In a typical WRFSI implementation the SRCROOT and INSTALLROOT share the same directory path name.)

3.1 Application Layout

The design of the WRF GUI is to present graphical tools to complete running of the WRFSI processes in a robust manner that makes sense to the users of the application. Instead of making users read much documentation to begin these tasks, the GUI provides partially and fully completed solutions for users to choose from, and then modify, to satisfy their goals. This allows users to bypass very complicated procedures, especially for those unfamiliar with scripting. Users can get useful results without otherwise knowing how to proceed from scratch.

The application layout consists of a 'tool selector' to the left of the window and a display of the selected tool to the right of the window. Always present in the GUI window are the standard menubar at the top of the window allowing the user to exit the application or query the help pages and release version information, found under File and Help buttons, respectively (upper left and right corners). We also find at the bottom of the GUI window a 'Hints & Information' area at times summarizing additional steps to be taken and other times processing that has been completed. See Figure 1.

There are several components to WRFSI and a specific sequence of processes that take place. The tool selector on the left side of the window assists the user in moving through these steps. Currently, there are three process steps to setting up WRFSI: Domain Selection, Initial Data, and Interpolate Data. A button and a corresponding editing tool represent each of these processes. Once a user selects a tool from the tool selector (by pressing a button), the process it represents is displayed and becomes editable in the right side of the window. These three tools are described in detail in sections 3.2 – 3.4.

Displaying a lot of information in a small space is a big challenge in a GUI application. The tabbed panel approach, where tabs are located across the top of each panel, is designed to address displaying lots of information in a small space. A tabbed panel for Domain Selection is shown in Figure 2. It allows the application to have many panels of information, but allows only one to be shown at a time. Another advantage of the tabbed panel approach is navigation through the localization tasks in a sequential fashion.

The over-all design approach includes suppressing detail and complexity until the user needs it. As such, some controls are deactivated and others are not displayed until they are relevant to the user accomplishing a task. Tool tips (small information boxes) are also displayed when the user places the cursor some parts of the application.

The GUI design was influenced by other similar applications that FSL evaluated during this development, including: the Air Force Weather Agency (AFWA) System GUI and the National Center for Atmospheric Research (NCAR/ARMY) MM5 GUI. (We'll need some references here).

3.2 Domain Selection Tool

The Domain Selection Tool has the following tabbed panels: Domain, Horizontal Grid, Vertical Grid, Localization Parameters, Initialization Controls and Localize Domain. The order in which the panels appear reflect the order in which the localization tasks need to be performed, starting from the left and proceeding to right. Below the panels are two control buttons: '<Back', and 'Next>' for the user to navigate through the tool. This guides users step by step through an otherwise complicated process with copious instructions and choices specified as needed.

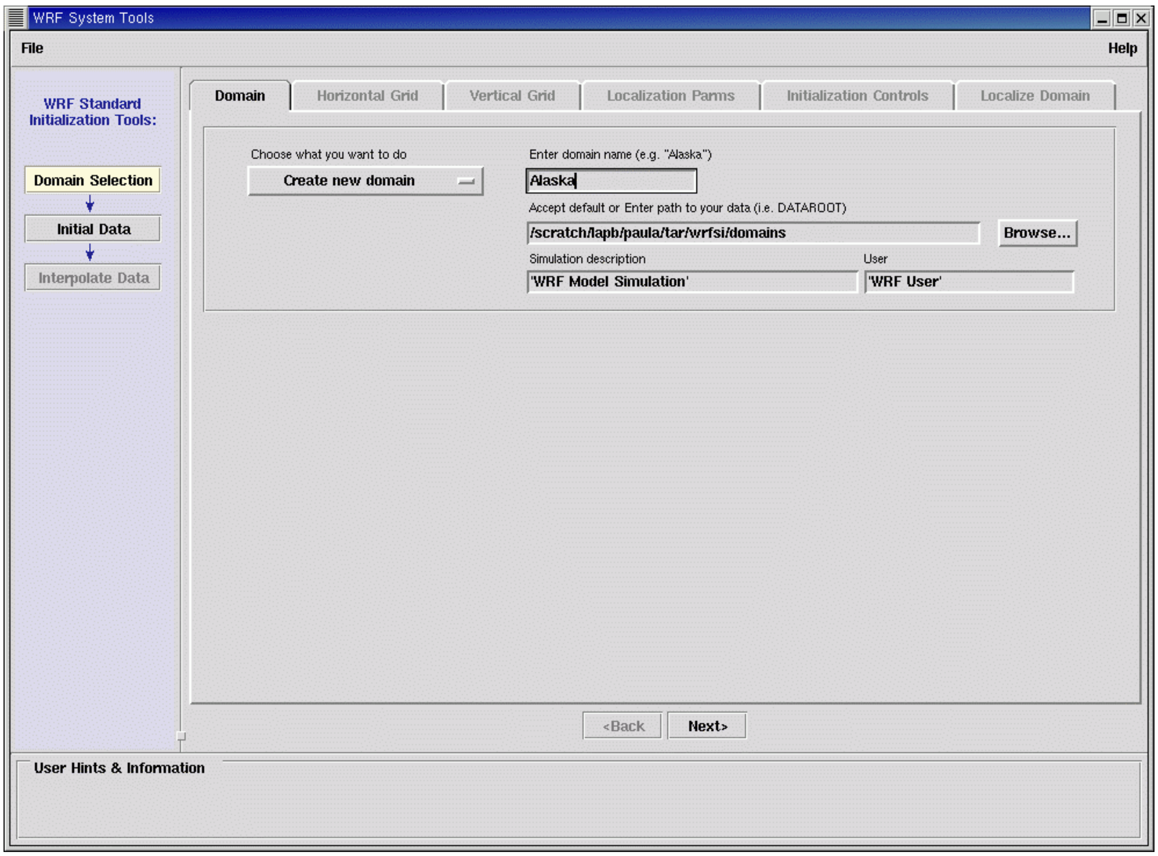

Figure 2: Display of Domain Selection Tool, showing the Domain panel.

3.2.1 Domain Panel

The Domain panel initiates the first task of the localization process; this is to declare a domain.

There are four domain modes available on the pull-down menu shown near the top of the panel allowing the user to choose to create, load, copy and delete domains. Once a domain is created, it can subsequently be edited, copied (to a case domain, Figure 3) or deleted depending on the wishes of the user and the mode they select. The GUI application includes a default domain, called 'default', to serve as an example of steps in the process of domain localization, though this particular domain cannot be edited or deleted only copied (then edited).

The Domain panel (Figures 2-4) has four entry fields: Domain name, Path to data or DATAROOT (with a file Browser widget), Simulation description and User description. The user can accept default values offered by the GUI or enter new values by typing them in. Typically users enter useful names for their domain; in this example an area encompassing Alaska will be localized and an appropriate name, Alaska, was entered as the domain name. A disabled entry signifies that an entry or button is unchangeable in this particular mode.

Once a domain has been localized (created), the GUI generates files that allow future reference to its DATAROOT. Because the DATAROOT directory contains the localization results and model output, the directory path entered must have ample disk space (or a disk partition). The environment variable called "MOAD_DATAROOT' is actually the DATAROOT plus the domain name. DATAROOT and MOAD_DATAROOT are synonymous in this paper.

We limit the discussion to new localizations; however, the GUI allows modification or review of existing domains. This is accomplished by selecting the option you want from the pull-down menu shown on the Domain panel.

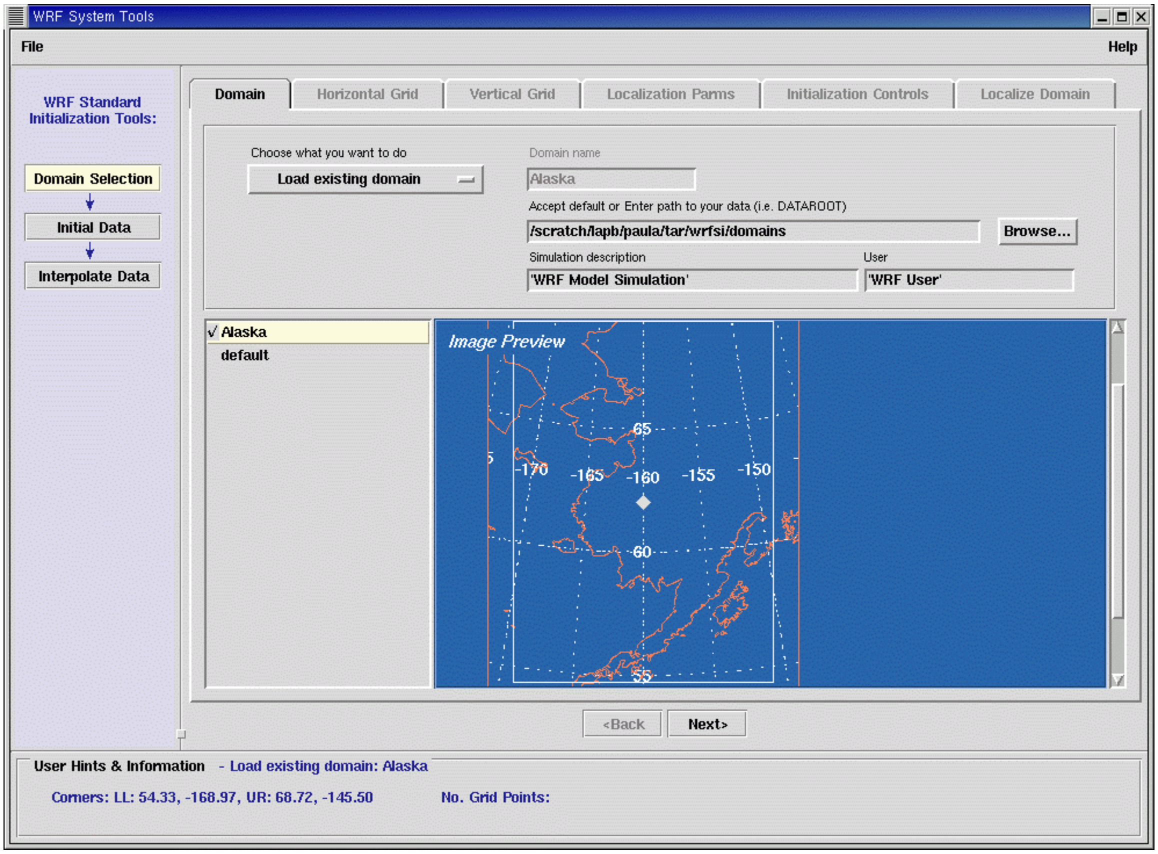

Figure 3: Display of Domain panel showing "Load existing domain" mode. Notice that there is a preview image of the domain.

The domain selection mode 'load' loads an existing domain into the GUI. Everything about the domain can be edited except for the domain name (and center point values), which is disabled.

The domain selection mode 'copy and edit' creates a new domain from an existing one. The automatic name for the new domain is generated from the original name adding a case number. For example, if Alaska is selected its domain name would be Alaska_001, but the user is free to enter a different domain name if they would like. There is a limit on the number same prefix named cases of 999.

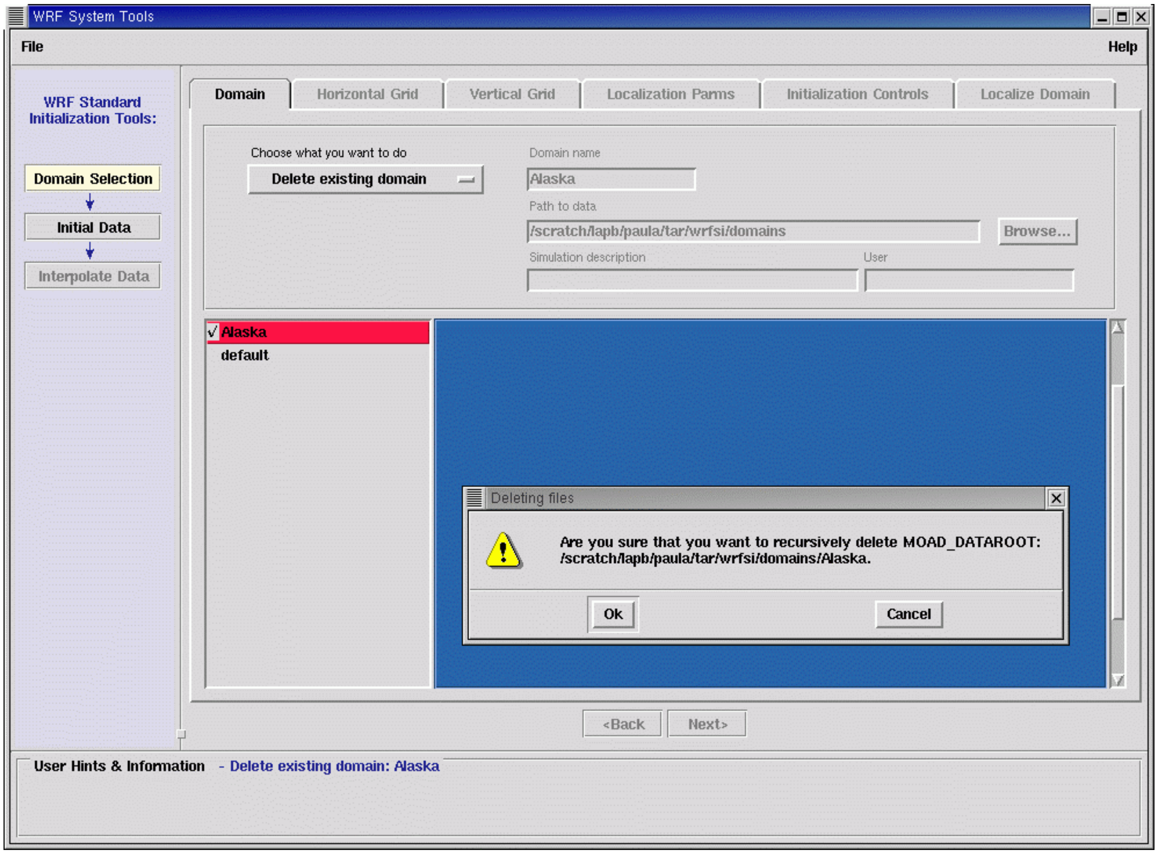

Figure 4: Display of Domain panel showing "Delete existing domain" mode.

The domain selection mode 'delete' highlights the selected domain selection in red and asks the user to confirm this action before a recursive remove is performed.

3.2.2 Horizontal Grid Panel

The Horizontal Grid panel (Figure 5) is comprised of two panels that contain tools allowing the user to define a new model domain by locating a box on a map (left panel) and editing initial map projection variables (right panel). The map window starts with a Cylindrical Equidistant projection map of the world centered at Latitude 0 and Longitude 0. The entire global map is not displayed, but the panel has a sliding scroll bar on the bottom and right sides to reposition it in the panel. The global map can be increased or decreased in size by pressing the appropriate zoom-in or zoom-out icon buttons just above the map.

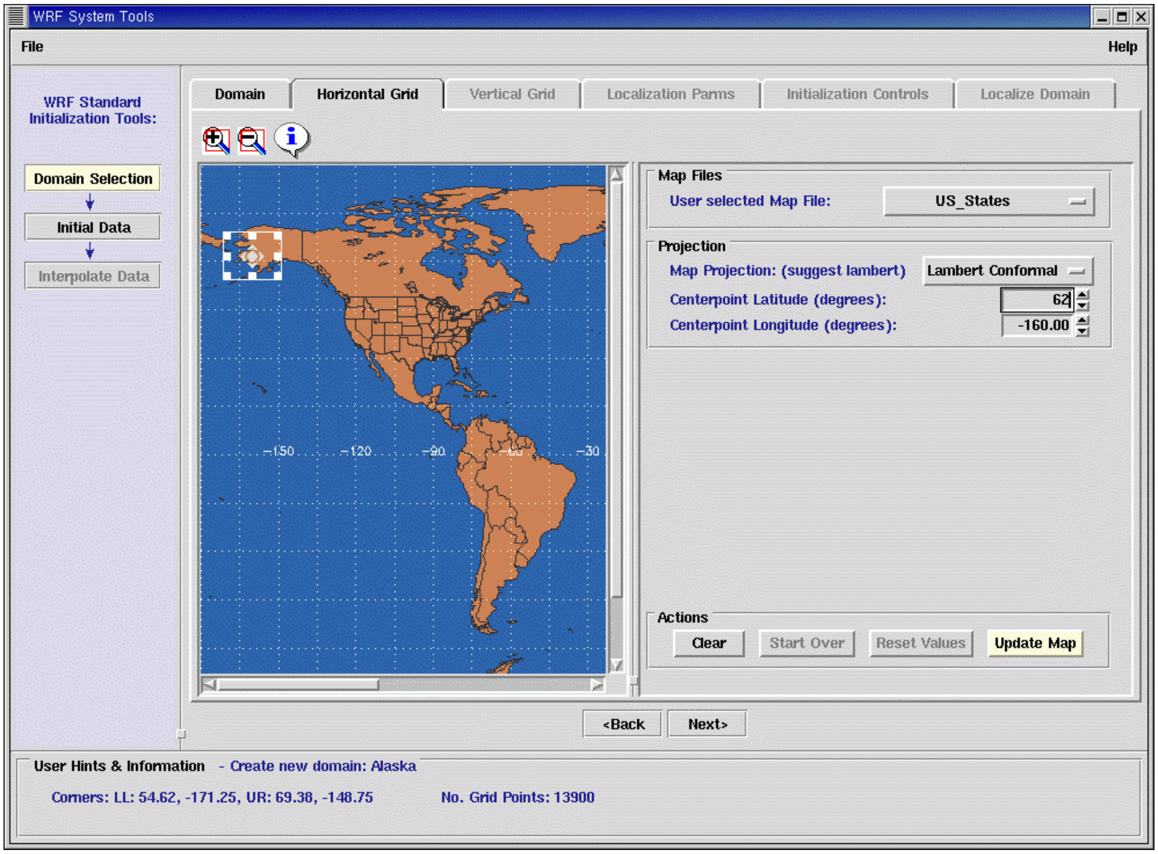

Figure 5: Display of Horizontal Grid panel.

A white domain bounding box will be drawn to indicate the location and extents of the domain as the user presses mouse button 1, on the global map, and drags the mouse to a new location on the map, as was done for the area over Alaska in Figure 5. The process of creating a domain box, then moving and resizing the box is described in detail by pressing the information icon button just above the map. Experimenting with the mouse button 1 and the 'Clear' button are also a good way to learn the functionality of creating a domain box and fine-tuning its size and location.

The top sections of the right control panel are used to set a background map, the projection type, and the center point values. As the domain box is moved or resized, the values in the control panel will update. Likewise, the user can manually edit the center point values in the right panel and the domain bounding box will update. NOTE: This is the only panel that allows modification of the center latitude and longitude and map projection.

The GUI has built in criteria to assist the user in selecting a projection based on domain's the center latitude value. The GUI label nearest the projection type pull-down menu suggested Lambert Conformal based on center latitude of 63 degrees, but Polar Stereographic was chosen instead (see Figure 5).

At the bottom of the control panel are action buttons. The 'Update Map' button is used to apply the projection parameters by drawing a new map. 'Start Over' will undo all updates but will leave the original domain bounding box. (This is an opportunity for the user to reenter center point latitude and longitude values.) 'Clear' resets all widgets and deletes the domain bounding box. Status information is sent to the Hints & Information panel at the bottom of the window. Once a domain box is created this panel displays both the lower left and upper right corner latitudes and longitudes of the domain box, and the total number of X, Y grid points in the selected domain.

There are warning dialog boxes that will appear with suggestions for the user if values are not selected prior to pressing the 'Update Map' or 'Next>' buttons. Notice that 'Update Map' is highlighted (turns yellow) to assist the user in the next step to take. Any changes to parameter values cause the 'Update Map' to be highlighted indicating the map image and the parameters are out of sync with each other. One of two things needs to happen, either press 'Update Map' to draw a new map, or press 'Reset Values' to return to the current maps parameters to the entry boxes.

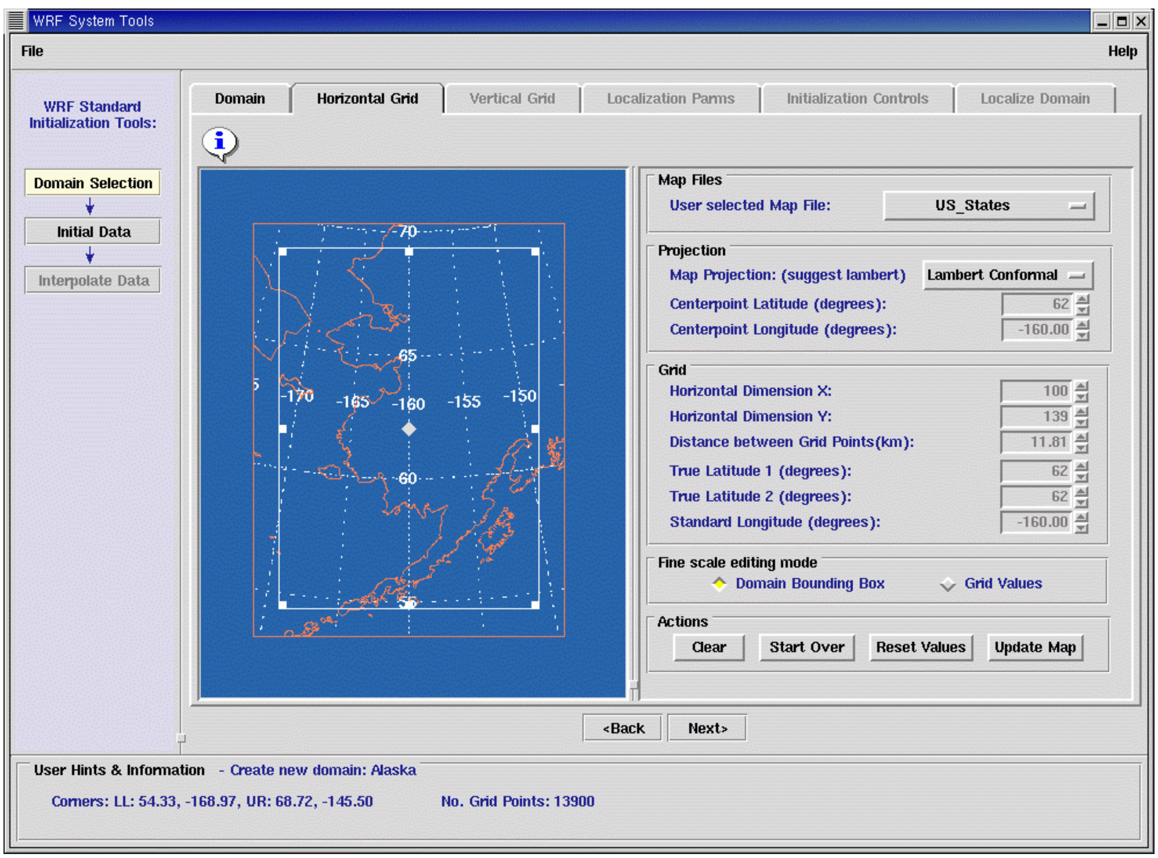

Figure 6: Display of Horizontal Grid panel following the user action of pressing 'Update Map'.

When the user presses 'Update Map' both left and right panels of the Horizontal Grid editor update as seen in Figure 6. The domain is now displayed in projection space (in this example, Polar Stereographic) based on user-selected information. You may notice that the lower left (LL) and upper right (UR) corner points have changed slightly due to the transformation from the previous map to the new gridded map in projection space. The new corner points are accurate. To obtain this gridded map, we first assume that ‘Horizontal Dimension X’ equals 100. The great circle distance between the east and west sides of the domain is computed which, in turn, is used to estimate ‘Grid Spacing’. Similarly, computing the north-south great circle distance and multiplying this by the grid spacing from the previous step determine ‘Horizontal Dimension Y’. Additional editing options are now available on the right control panel. These allow for more exact adjustments to the map. (The staggered grid when applied in gridgen_model will additionally affect the lower left and upper right corner points by slightly changing these values further.)

Most likely the user will continue to refine the domain. There are two modes to do fine scale editing, domain box and grid values:

1. The ‘Domain Box’ mode, when selected, allows the user to interactively resize the domain bounding box using the mouse to manipulate the white box in the left panel (Figure 6) while keeping the center point and grid spacing values fixed.

2. The ‘Grid Values’ mode, when selected, allows the user to enter values that will result in the domain bounding box adjusting (Figure 7). Note: the domain bounding box looses its tags to indicate to the user that the box is not currently interactive. When the user is satisfied with the domain’s map they proceed to the ‘Vertical Grid’ panel by pressing on ‘Next’.

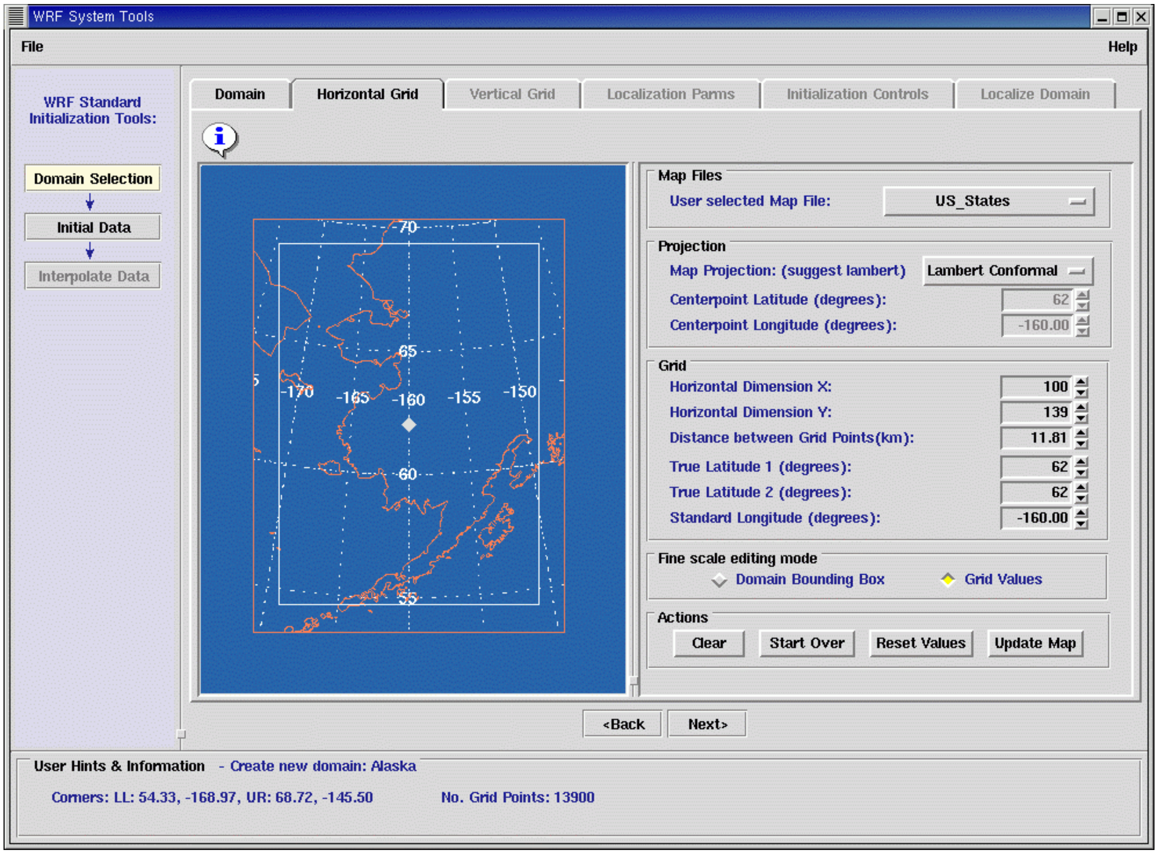

Figure 7: Display of Horizontal Grid panel with 'Grid Values' selected as the method to fine-tune the domain values.

3.2.3 Vertical Grid Panel

This panel (Figure 8) is comprised of two panels that contain tools that allow users to view vertical sigma levels (left side) and edit these sigma levels (right side). Included in the right control panel are several methods to edit the sigma levels. One method allows the user to select a vertical sigma level scheme and a corresponding desired number of sigma levels. The vertical display of sigma levels will update as these values are entered. In a second method, a sigma level file is input by using a file browser. A third method manually allows the user to edit the list of sigma levels displayed in a text window. Status information is sent to the Hints & Information panel at the bottom of the window. This contains the minimum and maximum sigma level distances, values affected surface pressure and temperature values and the pressure level at the top of the domain. Vertical sigma levels can be displayed in log pressure or not.

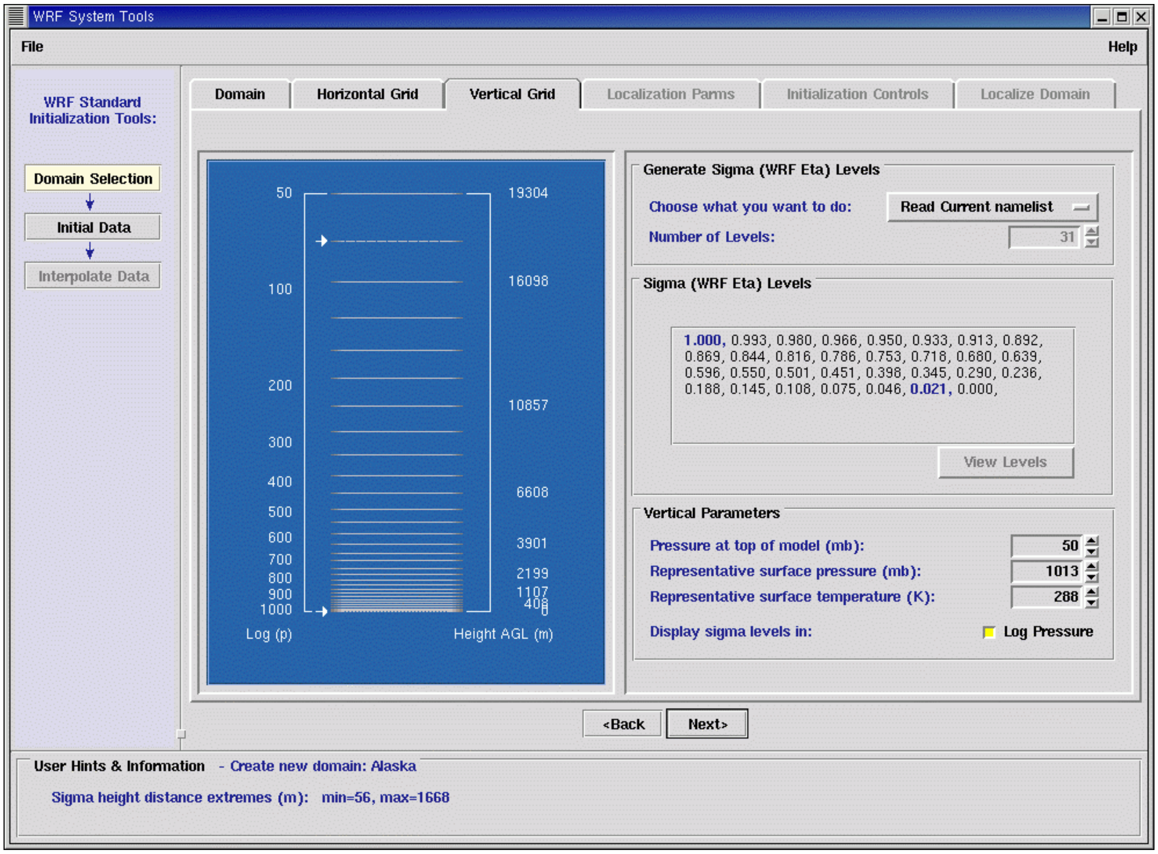

Figure 8: Display of Vertical grid panel.

3.2.4 Localization Parameters Panel

The WRFSI Fortran namelist file (called wrfsi.nl located in SRCROOT) is read in when the WRFSI GUI is started. From the beginning user selections and choices, affect the values of variables in the namelist. If the user is not satisfied with the values in the namelist, they should edit them. These entries and the paths to surfaces geography files are listed and are editable. Please refer to the WRFSI README file (Appendix A) to learn more about the geography data and how to acquire it.

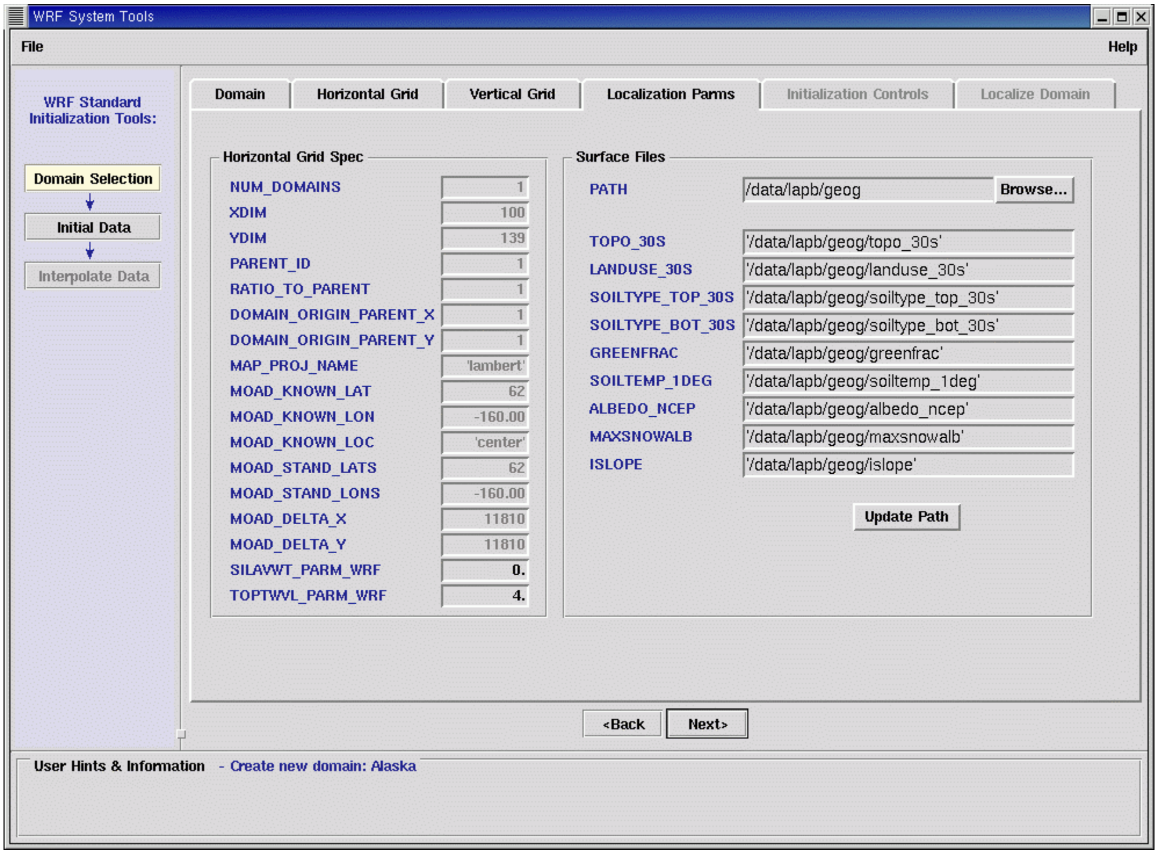

Figure 9: Display of Localization Parameters panel.

3.2.5 Initialization Controls Panel

This panel contains parameters relating to interpolating initial and lateral boundary condition files to the domain. These are written to the namelist. But are used by Perl scripts to prepare initial and lateral boundary file referred to and discussed in sections 3.3 Initial Data Tool and 3.4 Interpolate Data Tool. These entries and the standard initialization paths are listed and are editable.

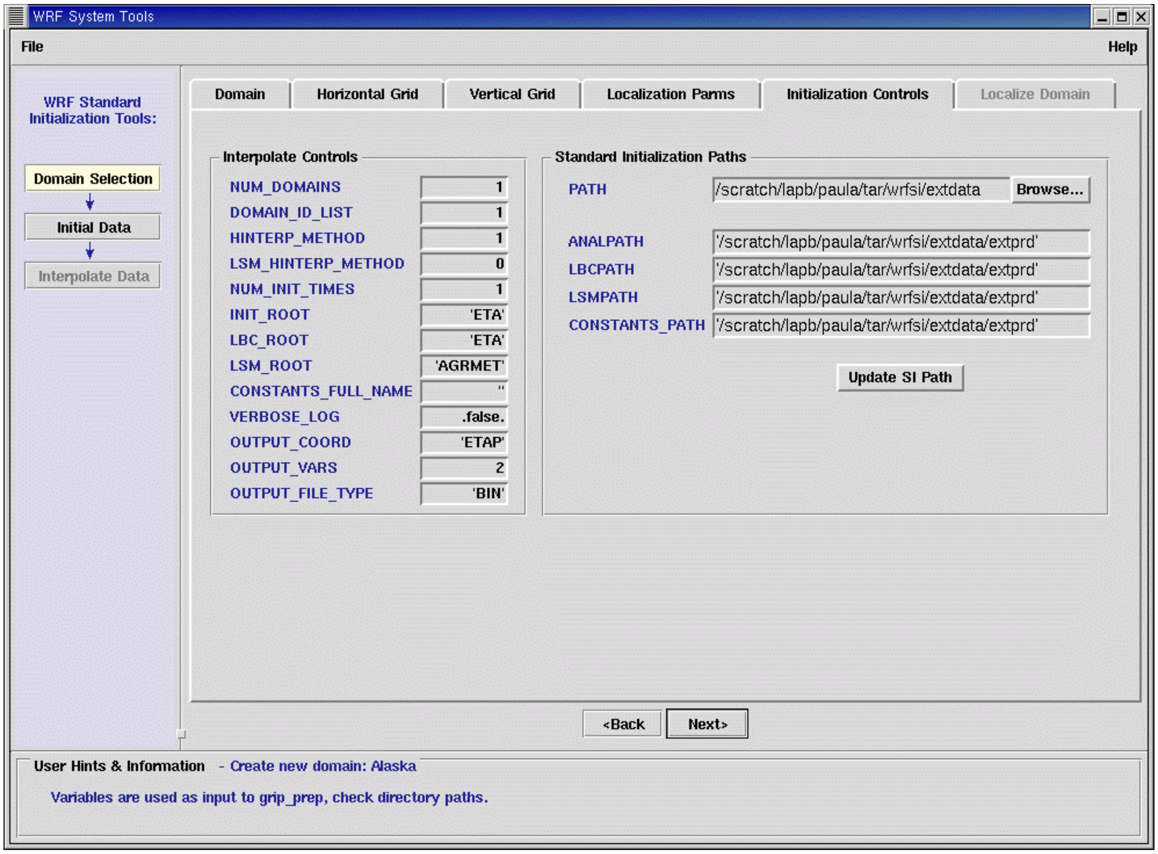

Figure 10: Display of Initialization Controls panel.

When the user is satisfied with the values in the previous and current Domain Selection panels, the next task is domain localization.

3.2.6 Localize Domain Panel

At this point the user is ready to run the Perl and Fortran programs to localize the domain. The window_domain_rt.pl command and all its command line arguments are shown in a text window. To run this process, the user presses ‘Localize’. Depending on the grid size and grid spacing of the configured domain, this process can take seconds to several minutes to complete. Output from the process, normally sent to the screen, is redirected to the same text window mentioned above.

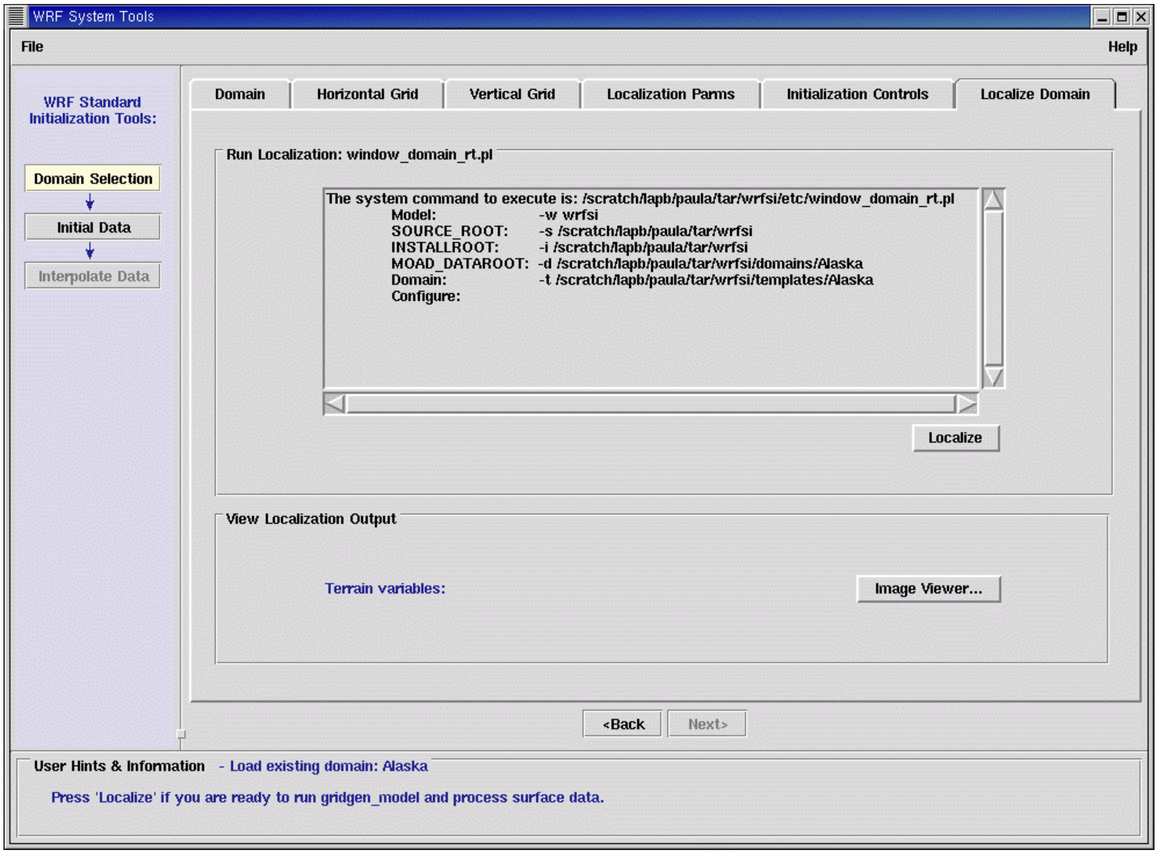

Figure 11: Display of Localize Domain panel. This user has NCAR Graphics NCL available and as such, they are able to view output from domain localization – with an image display tool dependant upon NCAR Graphics being installed.

The end product of this process is a defined and localize a three dimensional grid. Of specific interest is a netCDF (network Common Data Format) file in the DATAROOT called 'static.wrfsi' (hereafter, static file). The static file contains all non-varying information required for the WRF model, including terrain, land-use, latitudes and longitudes, map factor and several other quantities.

3.2.6.1 Graphics to Display Successful Localization

There is a capability built into the GUI to create and display graphics of the localization results. Currently this is only possible if NCAR Graphics 4.3.0 is installed with NCARG_ROOT and NCL_COMMAND properly set. Example images of graphics output from the Alaska localization include terrain (Figure 12) and land use (Figure 13). These were generated by IDL but are currently generated by NCAR Graphics NCL and are viewable by NCAR Graphics idt launched from the GUI. The full graphic output (ncarg meta) files include the following variables:

1. Terrain.
2. Terrain slope.
3. Land-use.
4. Soil type.
5. Greenness fraction.
6. Albedo.
7. Mean annual soil temperature.

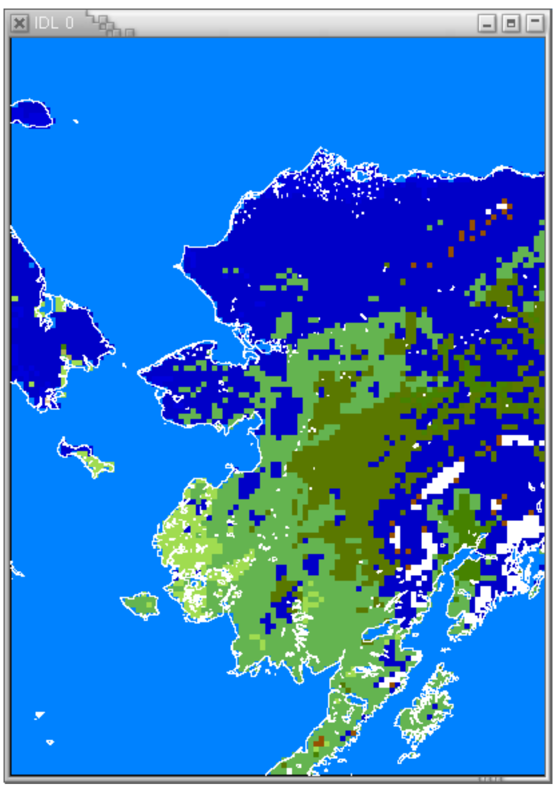

Figure 12: Image of land use for the Alaska domain localized in the example.

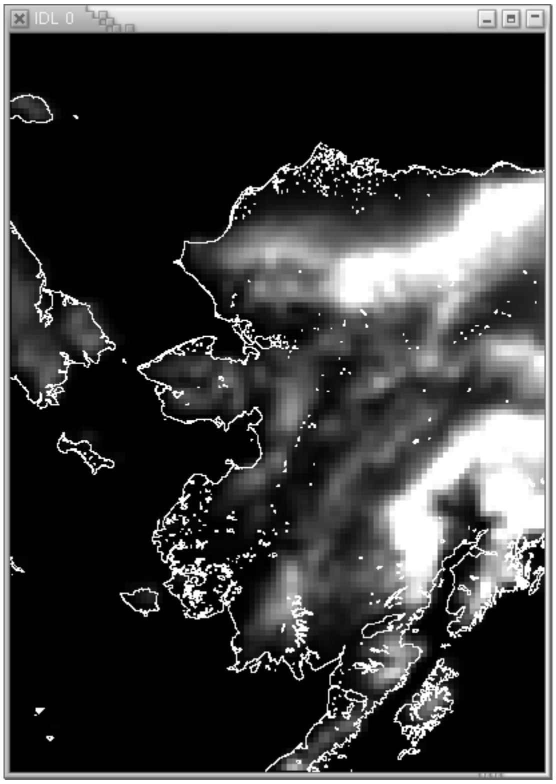

Figure 13: Image of terrain for the Alaska domain localized in the example.

3.3 Initial Data Tool

The Initial Data Tool performs the first of a two part task in providing the initial and lateral boundary condition data files. This section of the user interface runs a script (grib_prep.pl), which acquires background then uses specified directories to decode and store this data.

Please note that the values entered in the Domain Selection Tool in the 'Initialization Controls' panel (Section 3.2.5, Figure 10) need to correspond to choices made here, specifically, the values for ANALPATH and LBCPATH need to be considered.

3.3.1 Sources Panel

The *grib_prep Fortran executable* is responsible for decoding GRIB files. The grib_prep.nl namelist is a file that controls the execution of the grib_prep executable. This section of the GUI assists a user to edit their *grib_prep.nl file* for their system configuration.

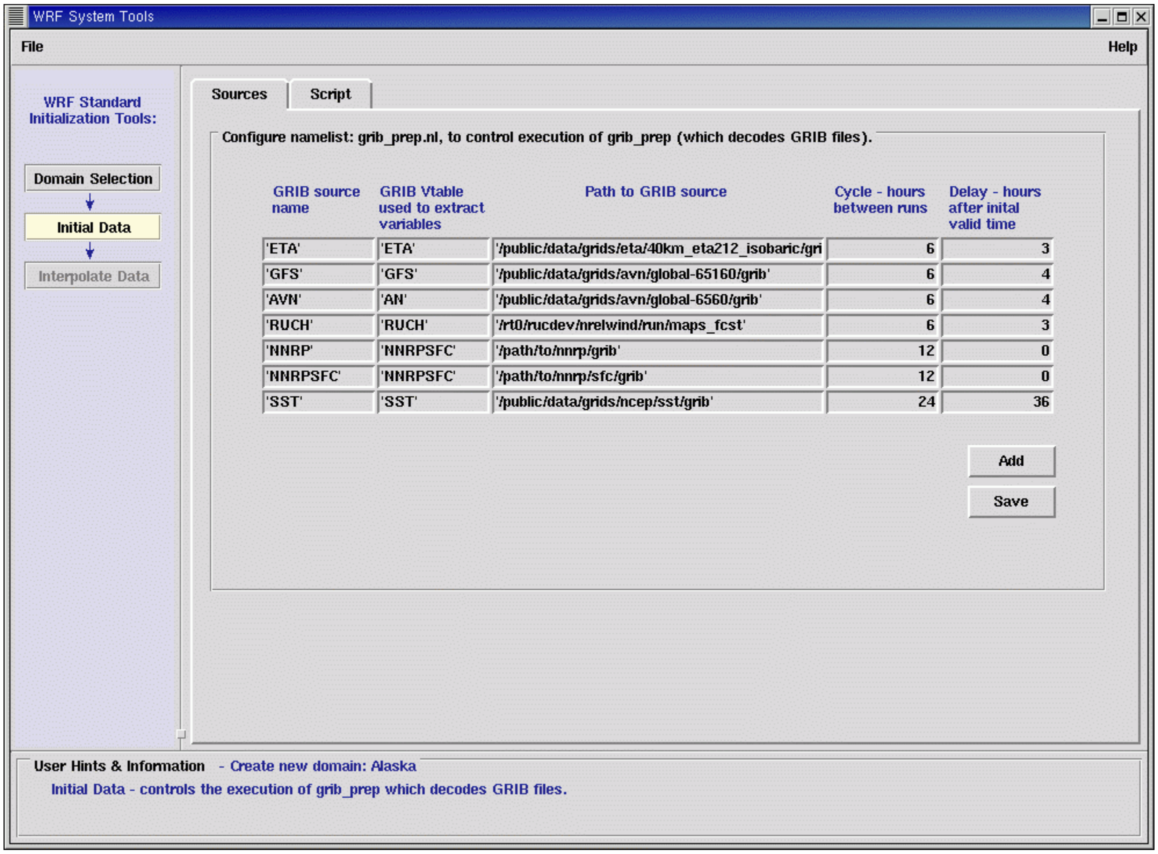

Figure 14: Display of Initial Data Tool - Sources panel.

As can be seen in the user interface (Figure 14) each entry line in the table of GRIB information is separated into five arguments that need to be edited for the users system configuration. These values can be edited or deleted by using the keyboard. An entire entry line can be left blank but for the underlying software to be successful, no values on a valid line should be left empty. To delete the GRIB source name, the first entry, is to invalidate (delete) the entire line.

'Add' creates a new blank line for user input. 'Save' both writes the grib_prep.nl namelist file but also updates the pull-down menu seen in the 'Script' panel (Figure 15).

3.3.2 Script Panel

This section of the GUI assists a user to configure their Perl *grib_prep.pl script*, which runs the Fortran executable grib_prep (which in turn reads the file grib_prep.nl).

First, the Script panel assists the user in selecting command line arguments to grib_prep.pl. By choosing a Data Source, for example ETA, an editable command is created in the text window shown at the bottom of the panel. A list of command line option switches is listed in the GUI resulting from running grib_prep.pl with the help option (grib_prep.pl -h). Second, the user is able to immediately run this command by pressing 'Run', or save the script to a file with 'Save'. (See Figure 15.) If Save is pressed the file will be located in INSTALLROOT/user_scripts available to be run at a later time, or used as input for a cron file, in the case of real-time runs.

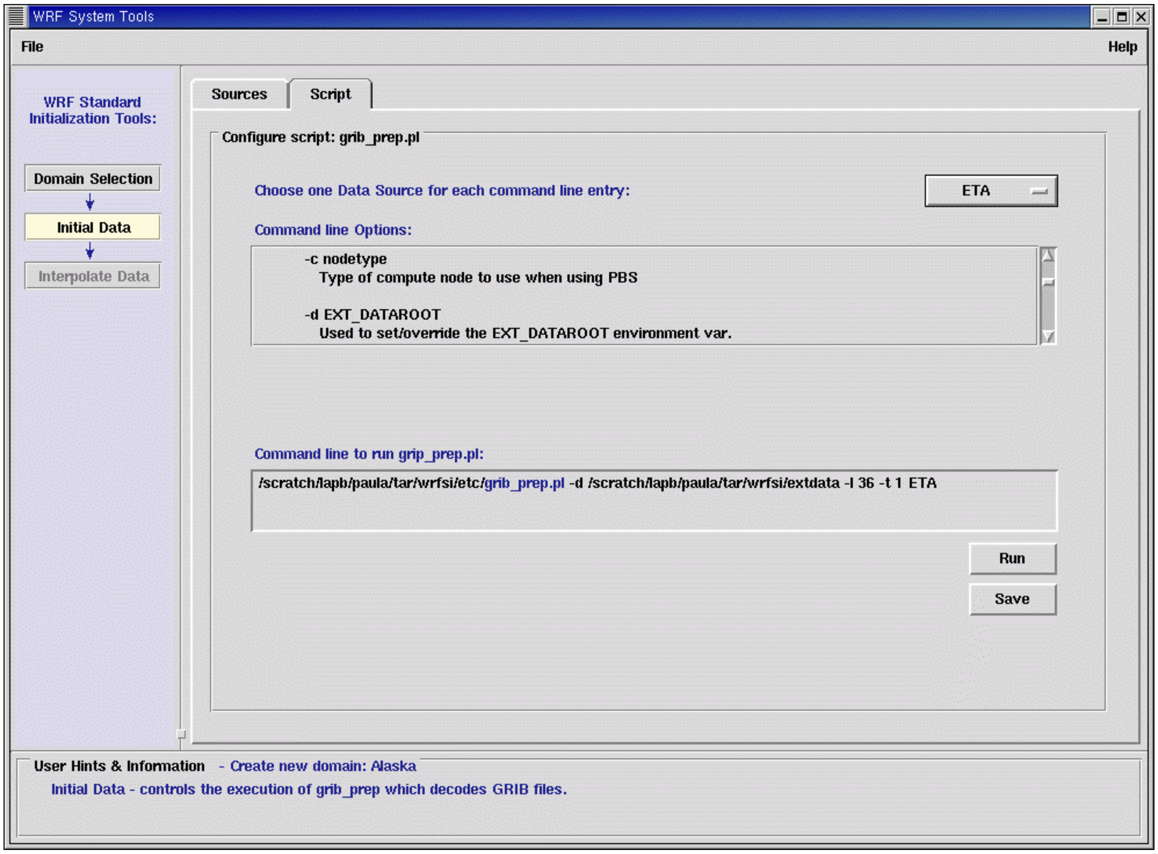

Figure 15: Display of Initial Data Tool - Script panel.

The grib_prep.pl script accepts only one data source argument at a time. Thus, an additional command line needs to be created for each data source to be processed. Please note that the values entered in the Domain Selection Tool in the 'Initialization Controls' panel (Section 3.2.5, Figure 10) need to correspond to choices made here, specifically the values for INIT_ROOT and LBC_ROOT need to be considered. A default of –t 1 instructs the GRIB data to create an hourly data file.

3.4 Interpolate Data Tool

The Interpolate Data Tool performs the second of two tasks in providing the initial and lateral boundary condition data files. This section of the user interface runs a script (wrfprep.pl), which interpolates background data to a specific domain. In fact, the button to invoke this tool is disabled (and appear gray) unless a localized domain is created or loaded via the Domain Selection Tool.

3.4.1 Script Panel

This section of the GUI assists a user to configure their *wrfprep.pl script*.

As can be seen in the user interface (Figure 16) this Script panel assists the user in selecting command line arguments to wrfprep.pl. A list of command line option switches is listed in the GUI resulting from running wrfprep.pl the help option (wrfprep.pl -h). The user is able to immediately run this command by pressing 'Run', or save the script to a file with 'Save'. If Save is pressed the file will be located in INSTALLROOT/user_scripts available to be run at a later time, or used as input for a cron file, in the case of real-time runs.

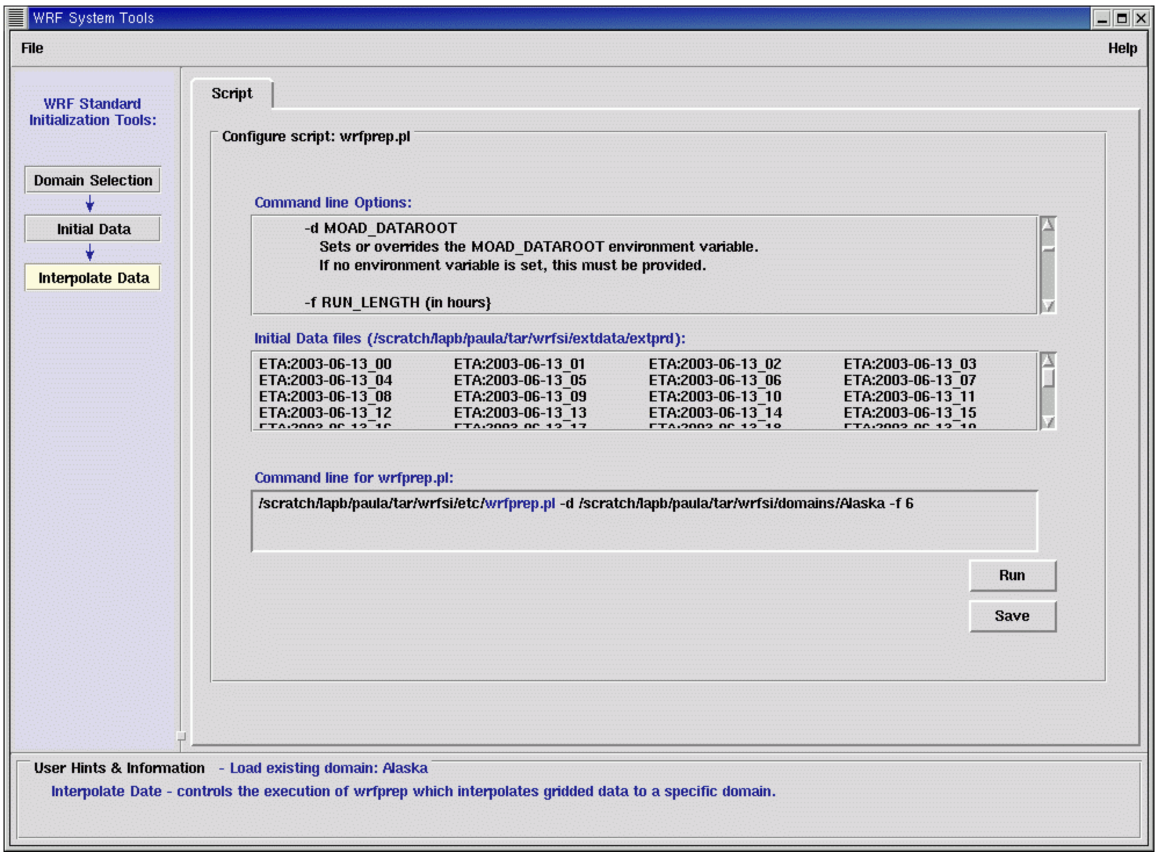

Figure 16: Display of Interpolate Data - Script panel.

The time option (-t) and forecast length (-f) switches used with grib_prep.pl directly relate to the option switches needed for wrfprep.pl to generate interpolated data. To illustrate, if you were to run grib_prep using -t 3 for the ETA files, you would get 12Z, 15Z, etc. If you were to run wrfprep.pl with no time arguments, it would run for the current hour (let's say, 16Z), which requires 16Z, 19Z, etc. for the ETA data. The result would be no interpolated files.

To generate desired file in this illustration, you could do one of many things:

1. Run grib_prep with -t 1, so you get hourly ETA output, thereby satisfying our ability to run the WRF for any hour.

-OR-

2. Use either -o or -s with wrfprep to start the model at a time for which you have some ETA data. For example, since you (for example) executed the scripts at 16Z, grib_prep determined that the 12Z ETA was the latest run, and created 12Z, 15Z, 18Z, ... output. You could run wrfprep.pl with -o -4 to say you want the initial hour 4 hours prior to the current hour (16-4 = 12Z). You could also run with -o -1 to use a 15Z initial time.

4.0 Future Work

4.1 Model Setup Tool

The GUI does not yet have a Model Setup tool allowing the user to set up the remaining part of WRFSI but we have plans to incorporate this functionality soon.

4.2 Acquiring Static Geographic Data

Localizing a WRF domain requires geography data files for 7 separate types of static land states information. These are listed in section 3.2.6.1 as the GUI generates the image displays after domain localization. More information is available in the README on the location, description and ***size*** of these static data sets.

5.0 Summary

The WRF GUI is not a finished product and there are plans for upgrades. On the other hand, the current GUI is capable of doing difficult first steps of numerical weather prediction set up in domain localization and generating interpolated initial and lateral boundary condition files. We envision the GUI to be capable of assisting the user with many other tasks associated with WRF system set up, running and evaluation.

6.0 References (incomplete at the moment)

LIDI02 Lidie, Steve, and Nancy Walsh. Mastering Perl/Tk. Sebastoopol (CA): O'Reilly & Associates, 2002.

JOHN00 Johnson, Jeff. GUI Bloopers: don'ts and do's for software developers and Web designers. San Diego: Academic Press, 2000.